Edinburgh Napier
UNIVERSITY

SECURITY & PRIVACY

Web Tech
SET08101

Simon Wells
s.wells@napier.ac.uk
http://www.simonwells.org

TL/DR

Edinburgh Napier UNIVERSITY

- The best way to build a secure & reliable web site/app?
  - Build Nothing.
  - Deploy Nowhere

Edinburgh Napier UNIVERSITY

AIMS

- At the end of this (sub-section) of the topic you will be able to discuss and evaluate:
  - Security Principles
  - Attacks
  - Role of Testing
  - Role of Engineering & Design
  - Privacy

Edinburgh Napier UNIVERSITY

PRINCIPLES

- Many overlapping themes: security, privacy, design, implementation, testing
- A process that we go through during the entire life-cycle of the software (not just development but also during deployment)
- Aim for triple-A security:

1. **Authentication**
2. **Authorisation**
3. **Auditing**

Edinburgh Napier UNIVERSITY

AUTHENTICATION

- Verifying that a person is a specific user
- Usually on the basis that they can provide security credentials:
  - username, password, answers to security questions, elements of shared secret, one-time-code (via app or SMS), fingerprint, iris scan, voice pattern
- For our purposes: **Generally checking credentials, e.g. username & password**

Edinburgh Napier UNIVERSITY

AUTHORISATION

- Matching particular user and
  - access to specific resources
    - permission to perform particular action
- Authorisation & authentication can overlap in complex ways as specific users may have a whole range of different capabilities
- For our purposes: which users can access which routes?

Edinburgh Napier UNIVERSITY

AUDITING

- Recording information about how your site functions
    - Failed login attempts
    - Suspicious requests
        - How do we decide what to audit?
- Helps recover after a breach
- Assuming the audit trail is reliable
- For our purposes:
    - Writing information about usage into logs from our web-app
    - NB. Usually your servers (e.g. NginX, Node.JS) also log data

Edinburgh Napier UNIVERSITY

SECURITY

- Drawing on general principles of software and computer systems security applied specifically to the Web (albeit with a large overlap to the Internet as well).

- Increased information held by websites about individuals due to increased social media (Web 2.0) usage in addition to increased use of the web as a place to do business & supply information services

- Leads to targeting of web sites, web applications, & underlying networks.

Edinburgh Napier UNIVERSITY

ATTACKS & VULNERABILITIES

- Most critical collated in the OWASP top 10
  - Updated annually (2017)
- Cross-Site Scripting (XSS)
- Cross-Site Request Forgery
- SQL/Code Injection
- Path Disclosure
- (Distributed) Denial of Service (DNS)
- Arbitrary Code Execution
- Arbitrary/Local/Remote File Inclusion
- Phishing

Edinburgh Napier UNIVERSITY

BEST PRACTISE

- You don't have to be security researcher to:
- **Take an active interest in security & privacy**
- Every developer should be considering their own skills & practices during design, implementation, & deployment
- OWASP supplies cheat sheets for:
  - API Development
  - Web Application Development

Edinburgh Napier UNIVERSITY

CONTENT SECURITY POLICY

- Security standard designed to prevent cross-site scripting, clickjacking, & code injection attacks that result in execution of malicious content in the trusted web page context.

- A candidate recommendation form the W3C working Group on Web Application Security.

- Standard method for website owners to declare approved origins of content (e.g. JavaScript, CSS, HTML frames, web workers, fonts, images, embeddable objects such as Java applets, ActiveX, audio and video files) that browsers should be allowed to load on that website

- Uses **Content-Security-Policy** header

- If header present in server response then client enforces a declarative whitelist policy

Edinburgh Napier UNIVERSITY

TESTING

- White Box testing (e.g. automated/static source code analysis/evaluation)
- Black Box testing (e.g. automated vulnerability scanners)
- Fuzzing
- Password testing (ensuring that passwords are "up to standard" - but what counts as a secure password?)
- Note: There are also other forms of testing, e.g. usability & performance, which have different aims
- **Real software development involves lots of testing**

Edinburgh Napier UNIVERSITY

WHITE BOX TESTING

- Source code level
- A method that looks at how software works internally then determines which tests to perform to determine that it is working to specification
- Tester chooses inputs in order to exercise the internals and identify expected & unexpected results
- Often happens at unit level but can be applied at integration & systems levels as well.

Edinburgh Napier UNIVERSITY

LEVELS OF TEST

1. Unit Testing - small individual/isolated units of code at commit/ requirement level. Possibly using test harness.

2. Integration Testing - Interactions between interfaces when units are combined & work together in a more open environment

3. Regression Testing - Used to ensure that new code hasn't broken existing code

4. System Testing - Exercising the software when all components are assembled together

Edinburgh Napier
UNIVERSITY

DESIGNING WHITE BOX TESTS

- Idea is to get coverage of the code covering multiple overlapping facets.
- Some of these overlap to some degree:
  - Control flow testing
  - Data flow testing
  - Branch testing
  - Statement testing
  - Decision coverage
  - Path testing

Edinburgh Napier UNIVERSITY

BLACK BOX TESTING

- Testing that examines the functionality of software in the absence of knowledge about how it is implemented.
- Tester is aware of what the software is designed to do but not how it does it
- Write test cases based on *Design* (specification & requirements)
- Select valid & invalid inputs, determine the correct output (can use an *oracle* to provide correct data)
- Usually higher level than white-box testing (integration/system level)
- Doesn't require programming knowledge

Edinburgh Napier UNIVERSITY

BLACK BOX PROCESS

1. Examine requirements & system specification
2. Choose valid inputs (test positive scenarios) then invalid inputs (negative test scenarios)
3. Determine expected outputs for the inputs
4. Construct test cases
5. Execute test cases - using either automated systems or humans following scripts
6. Compare expected to actual outputs
7. Fix defects, re-test

Edinburgh Napier UNIVERSITY

BLACK BOX TESTS

- Decision Table testing
- Input pair testing
- Error guessing
- State Transition testing
- Use case testing
- User story testing
- Domain analysis

Edinburgh Napier UNIVERSITY

FUZZING

- Automated generation of (invalid, unexpected, random) semi-structured inputs to a program to see how it behaves:
    - It might be fine
    - it might crash
    - it might cause data corruption
    - it might expose data an undesirable way
- Semi-structured - Not just random data. Things that aren't what we expected but aren't outright rejected by our input parser. Things that can expose edge/corner/boundary conditions which might not have been properly dealt with.
- Consider what should be allow for a *username* or for an *email address* input

Edinburgh Napier UNIVERSITY

USER DATA

- Pay attention to user inputs
- Particularly inputs that cross **trust boundaries**
  - e.g. user input from a form, upload, or even the address bar
- Code related to these kinds of situation carry more risk than code that runs internally
  - e.g. parsing an internal configuration file at startup that is only available to privileged users
- Not saying that there aren't internal threats but there is a limited amount of time/effort available and external threats are quantifiably different to internal threats (although internal threats can be both worse and more insidious than an outsider)

Edinburgh Napier UNIVERSITY

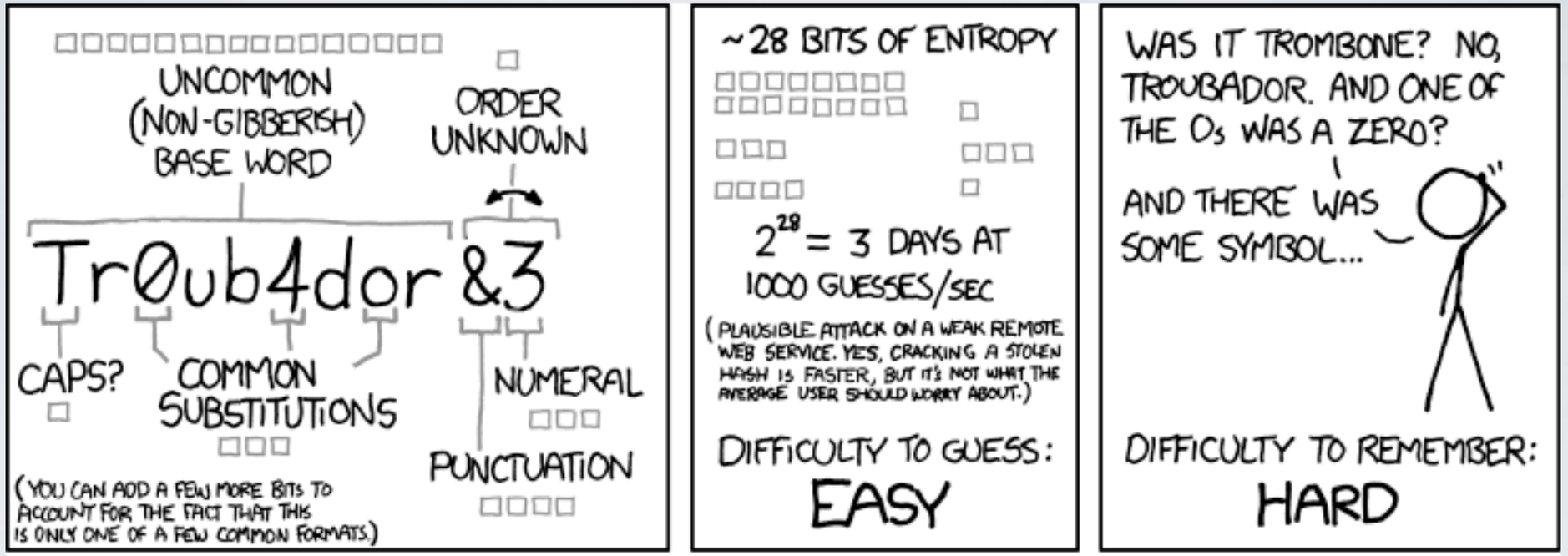
PASSWORDS

- String of characters used for **authentication**
- Frequent weakness of systems
- Many collections of passwords are traded online:
  - https://haveibeenpwned.com/

FACTORS AFFECTING SYSTEMS THAT USE PASSWORDS

Edinburgh Napier UNIVERSITY

- Rate at which attackers can try to guess passwords
  - NB. Access to password database dump
- Number of guesses
- Form of stored passwords
- Transmission of password (simple, encrypted, challenge-response, zero-knowledge)
- Changing passwords & password reuse
- Longevity & password ageing
- Shared versus individual passwords


UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor&3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS:
EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER:
HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS:
HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER:
YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

TESTING: WEB APPS

- via OWasp:
  - Information Gathering
  - Configuration Management
  - Secure Transmission
  - Authentication
  - Session Management
  - Authorization
  - Cryptography
  - Data Validation
  - Denial of Service
  - Specific Risky Functionality
  - Error Handling

TESTING: WEB SERVICES

- via OWasp:
  - Transport Confidentiality
  - Server Authentication
  - User Authentication
  - Transport Encoding
  - Message Integrity
  - Message Confidentiality
  - Authorization
  - Schema Validation
  - Content Validation
  - Output Encoding
  - Virus Protection
  - Message Size
  - Availability:
    - Message Throughput
    - XML Denial of Service Protection
  - Endpoint Security Profile

Edinburgh Napier UNIVERSITY

PRIVACY

- Ability of an individual of group to seclude themselves or their information, and thereby selectively express themselves

- Varies by culture & by individual

- Secrecy & privacy:

“When I withhold information, it is privacy; when you withhold information, it is secrecy”

- Overlaps with security issues

- Increasingly the subject of public policy & law

PRIVACY CONCEPTS

1. Right to be let alone
2. Option to limit access by others to one's personal information
3. Option to conceal information from others
4. Control over use by others of one's information
5. States of privacy:

   (a) Solitude (physical separation), (b) intimacy, (c) anonymity, (d) reserve (psychological barrier)
6. Personhood & autonomy
7. Self-identity & personal growth
8. Protection of intimate relationships

Edinburgh Napier UNIVERSITY

PRIVACY BY DESIGN

- Approach to engineering social & computational systems
- Take privacy into account throughout entire engineering process
- Foundational Principles:
    1. Proactive not reactive
    2. Privacy as default setting
    3. Privacy embedded into design
    4. Full functionality
    5. End-to-end (full lifecycle) security
    6. Openness - Visibility & transparency
    7. User-centric design

Edinburgh Napier UNIVERSITY

SUMMARY

- Security Principles
- Attacks
- Role of Testing
- Role of Engineering & Design
- Privacy

Edinburgh Napier UNIVERSITY

NEXT

- Wrapping up the module